Intelligent Autonomous Systems Project 2: Gesture Recognition

Ethan Glaser

eg492

I. Introduction and Problem Formulation

This project focuses on creating an algorithm to classify the trajectory of several motion patterns using IMU accelerometer and gyroscope readings over time. 6 different paths of motion (circle, infinity, eight, beat3, beat4, and wave) have been repeated over several iterations with IMU data captured consistently throughout the motion to create a dataset representing each of these movement patterns. The goal of this project is to use these samples to train Hidden Markov Models that can determine the likelihood of new IMU readings and select the most probable motion using the HMMs.

II. Technical Approach

The primarily technical concept behind the algorithm for determining the pattern of motion is the Hidden Markov Model, however there is significant preprocessing that must be completed on the data leading up to the HMM usage. The first step is to classify each IMU reading using clustering. KMeans was implemented with a variable number of clusters, with the cluster centroids determined using every single file in the training data after normalization was performed on the data (subtracting mean and dividing by standard deviation for each column). The mean and standard deviation for each feature and the centroids can then be saved to numpy files so that this process doesn't have to be repeated (unless trying a new number of clusters). Initial visualization was done using 50 clusters, and Figures 1-4 show the results of the clustering on some of the provided files, with a quick visual scan indicating that it is relatively easy to identify the motion solely based on the clustered datapoints over time.

Once the data is clustered, the training of Hidden Markov Models can be implemented. This process was completed following the outline provided in the course slides, the Rabiner HMM paper, and the Baum Welch Wikipedia page. The first step is to initialize A (probability of transitioning from one hidden state to another), B (probability of being at hidden state for a given observation), and pi (probability of starting at a hidden state) to ensure that respective rows and columns sum to 1. Next, repeat the process of updating Alpha and Beta (probabilities of certain order of hidden states) using the current value of A and B and the most recently determined values for Alpha and Beta, then update A and B using the recently updated Alpha and Beta matrices. At a given iteration in this process, the log likelihood for the A, B, pi combination can be observed by the final calculated Alpha and Beta values. Figures 5-10 show the log likelihood calculations as a function of training epochs for each of the 6 motion patterns, each of which is strictly negative and decreasing in magnitude.

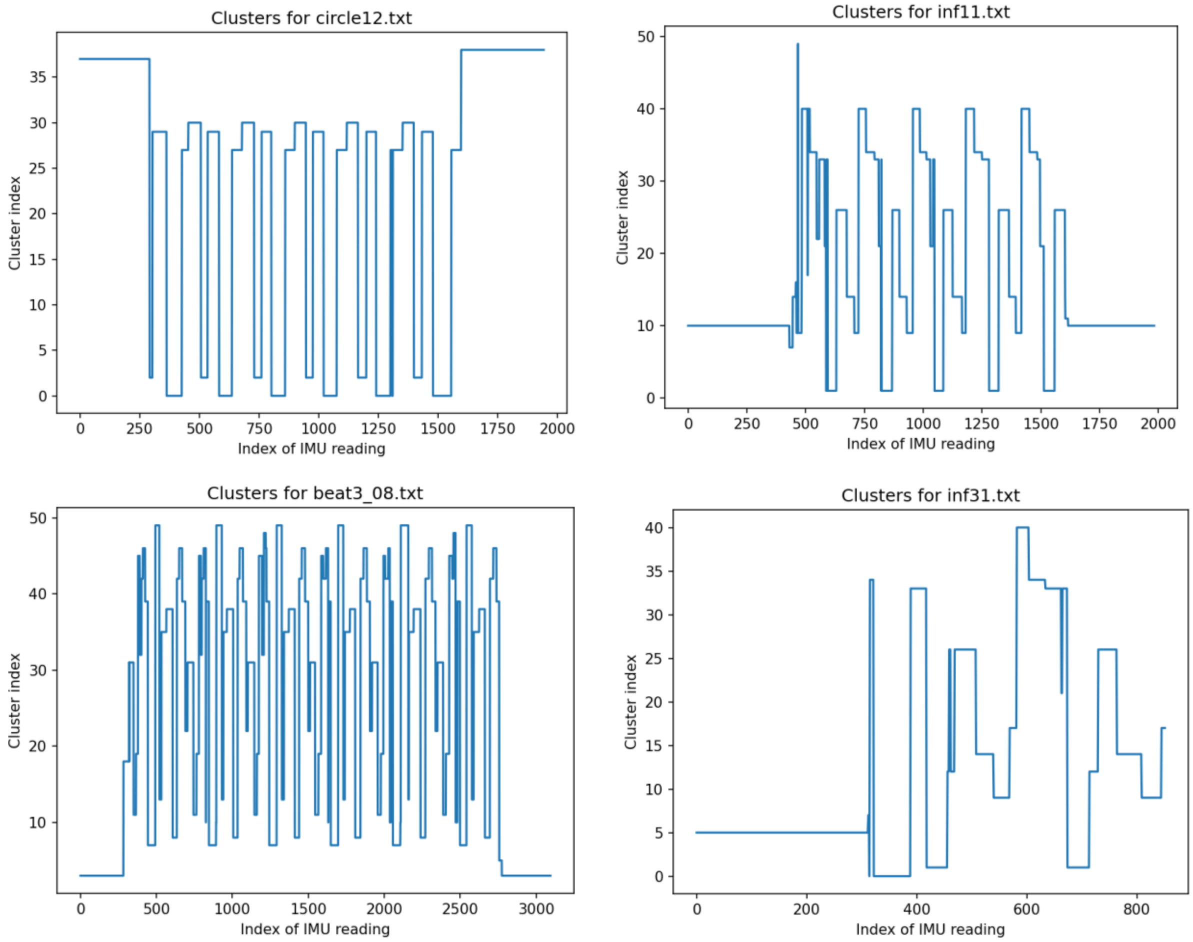
Once training was complete, testing could be conducted on the extended training set using the A, B, and pi matrices determined for each of the 6 motion patterns. Testing followed a similar process as training just without initializing and updated A, B, and pi (just calculating alpha and beta). Each test instance was tested against each of the 6 Hidden Markov Models and the lowest log likelihood was then determined to be the predicted motion for that sample. There were 3 primary hyperparameters involved with training and testing – number of training epochs, number of clusters, and number of hidden states. The number of training epochs was not tuned since the log likelihood plots indicated that going longer than 10 epochs did not improve performance. Therefore, just the number of clusters and number of hidden states were tuned using a grid search, with expected number of clusters being between 50 and 100 and expected number of hidden states being between 10 and 20, the hyperparameter tuning did not lead to very unexpected results and the selected number of clusters was 50 and number of hidden states was 15, and these parameters achieve 100% accuracy (6/6) on the extended training set as well as 93% accuracy (28/30 with the last 2 being the second choice) on the original training set.

III. Results and Discussion

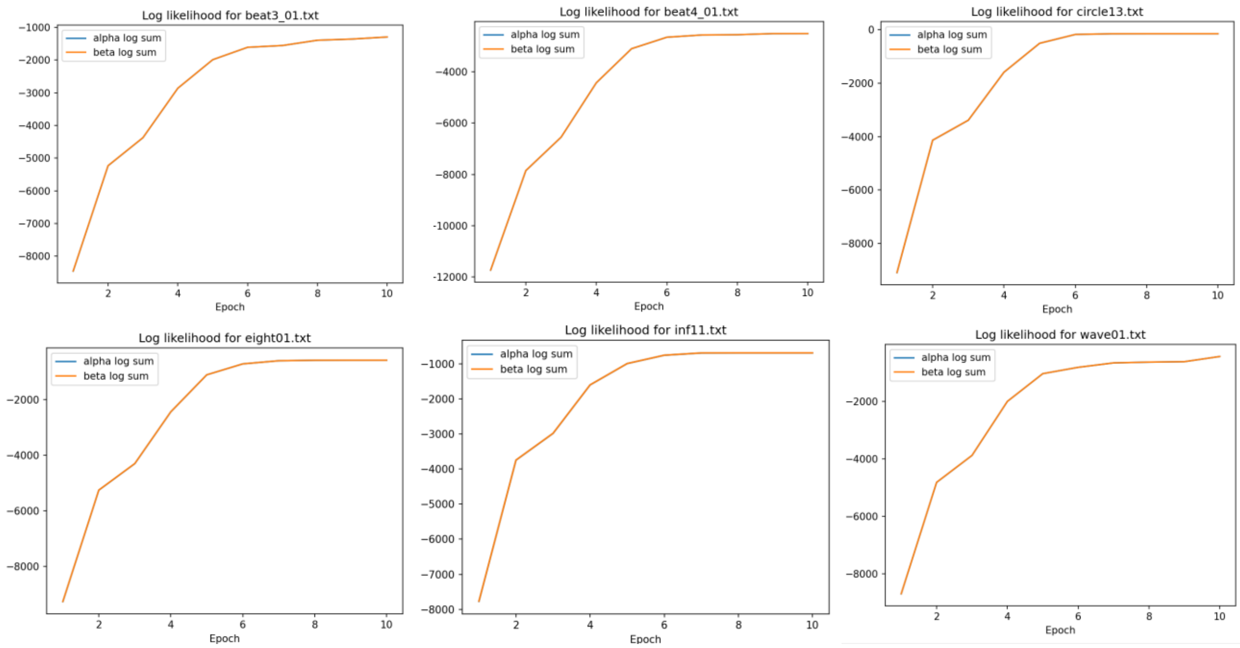
On initial submission, the process of creating A and B matrices followed the algorithm directly, which led to many entries defaulting to 0 over time. This led to some difficulties with testing on patterns that were slightly different than those observed in the training. As a result, many log likelihood values hit nan values due to division by 0 errors. The initial submission evaluations were at least able to identify one or two non-nan values for each test case, and the original predictions are shown in Figure 11. Updates were made to the training process after the initial submission to be more forgiving for cases that hadn't been observed in training – specifically a small value (1e-8) was added to each entry in the A and B matrices each iteration to allow for some wiggle room in the log likelihood calculation. This led to a much cleaner result, with no cases erroring out. The final results are shown in Figure 12 with more detail included in Figure 13, but the table below shows the final predictions of the unknown test cases. Figure 3 provides details that can indicate how confident each of the predictions is – with most Beat3 and Beat4 predictions being relatively close in magnitude as well as Inf and Eight, which makes sense considering the large amount of overlap in the motions.

File name	First Choice	Second Choice	Third Choice
test1.txt	Wave	Eight	Inf
test2.txt	Beat3	Beat4	Wave
test3txt	Inf	Eight	Wave
test4.txt	Beat4	Beat3	Wave
test5.txt	Circle	Beat4	Beat3
test6.txt	Inf	Eight	Beat3
test7.txt	Eight	Inf	Beat3
test8.txt	Beat3	Beat4	Wave

Table 1: Final Predictions



Figures 1-4: Cluster indices for 3 training files (circle, infinity, beat3) and single sample of infinity. Visually, the infinity sample closely matches that of the training file, especially compared to the other two.



Figures 5-10: Log likelihood of training for 1 file of each of the 6 motion patterns

```
Prediction for test1.txt: wave  (second choice:      )
Prediction for test2.txt: beat3  (second choice: beat4)
Prediction for test3.txt: inf  (second choice:      )
Prediction for test4.txt: beat3  (second choice:      )
Prediction for test5.txt: circle  (second choice:      )
Prediction for test6.txt: inf  (second choice:      )
Prediction for test7.txt: eight  (second choice:      )
Prediction for test8.txt: beat3  (second choice: beat4)
```

Figure 11: Predictions from originally submitted model

```
Prediction for test1.txt: wave  (second choice: eight, third choice: inf)
Prediction for test2.txt: beat3  (second choice: beat4, third choice: wave)
Prediction for test3.txt: inf  (second choice: eight, third choice: wave)
Prediction for test4.txt: beat4  (second choice: beat3, third choice: wave)
Prediction for test5.txt: circle  (second choice: beat4, third choice: beat3)
Prediction for test6.txt: inf  (second choice: eight, third choice: beat3)
Prediction for test7.txt: eight  (second choice: inf, third choice: wave)
Prediction for test8.txt: beat3  (second choice: beat4, third choice: wave)
```

Figure 12: Final predictions with updated model

```
[('wave', -536.7730862998878), ('eight', -4182.740311230445), ('inf', -5324.034128155308), ('beat4', -6306.583189517745), ('circle', -6378.099047649438), ('beat3', -6944.731669571486)]
Prediction for test1.txt: wave  (second choice: eight, third choice: inf)
[('beat3', -1167.940061945966), ('beat4', -1323.1204950024287), ('wave', -7305.025038549116), ('inf', -8321.285251578207), ('eight', -9943.5204944448), ('circle', -14984.711760619111)]
Prediction for test2.txt: beat3  (second choice: beat4, third choice: wave)
[('inf', -1169.439954969344), ('eight', -1891.6195256525953), ('wave', -9254.633555161625), ('beat4', -10331.257469717839), ('beat3', -10345.31004114715), ('circle', -15621.10930734661)]
Prediction for test3.txt: inf  (second choice: eight, third choice: wave)
[('beat4', -1329.199657117349), ('beat3', -1526.8757115443925), ('wave', -6284.431036477647), ('inf', -7280.6887820182765), ('eight', -12997.764562909832), ('circle', -13583.10757155093)]
Prediction for test4.txt: beat4  (second choice: beat3, third choice: wave)
[('circle', -848.6348889760203), ('beat4', -5273.90097328349), ('beat3', -6265.932310347279), ('wave', -6893.522873100538), ('inf', -12160.24478452393), ('eight', -12692.191578162341)]
Prediction for test5.txt: circle  (second choice: beat4, third choice: beat3)
[('inf', -1447.7634324941557), ('eight', -2165.3728509047814), ('beat3', -10694.869201939293), ('wave', -14847.36170006219), ('beat4', -15483.200358526232), ('circle', -15886.270491031162)]
Prediction for test6.txt: inf  (second choice: eight, third choice: beat3)
[('eight', -1218.60406715861), ('inf', -1519.980285020048), ('beat3', -8749.51510555003), ('wave', -9591.863438527003), ('beat4', -10134.90177329768), ('circle', -14036.89295296395)]
Prediction for test7.txt: eight  (second choice: inf, third choice: beat3)
[('beat3', -1116.77257964024), ('beat4', -1267.664866155611), ('wave', -7011.975052540908), ('inf', -7937.6263525945915), ('eight', -9450.273865635689), ('circle', -14484.39257276733)]
Prediction for test8.txt: beat3  (second choice: beat4, third choice: wave)
```

Figure 13: Final predictions with log likelihood for each motion computed